Scientific Reports

https://doi.org/10.1038/s41598-026-64187-3

Article in Press

Reducing circuit resources in Grover's algorithm via constraint-aware initialization

Received: 29 January 2026

Accepted: 23 July 2026

Published online: 29 July 2026

Cite this article as: Bae E., Shin J. & Choi M. Reducing circuit resources in Grover's algorithm via constraint-aware initialization. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-64187-3

Eunok Bae, Jeonghyeon Shin & Minjin Choi

We are providing an unedited version of this manuscript to give early access to its findings. Before final publication, the manuscript will undergo further editing. Please note there may be errors present which affect the content, and all legal disclaimers apply.

If this paper is publishing under a Transparent Peer Review model then Peer Review reports will publish with the final article.

ARTICLE IN PRESS

© The Author(s) 2026. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

ARTICLE IN PRESS

Reducing Circuit Resources in Grover's Algorithm via Constraint-Aware Initialization

Eunok Bae[1], **Jeonghyeon Shin**[2,3], **and Minjin Choi**[4,*]

[1]Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea
[2]Center for Quantum Technology, Korea Institute of Science and Technology, Seoul 02792, Republic of Korea
[3]Department of Mathematics and Research Institute for Basic Sciences, Kyung Hee University, Seoul 02447, Republic of Korea
[4]Center for Quantum Information R&D, Korea Institute of Science and Technology Information, Daejeon 34141, Republic of Korea
[*]mathcmj89@gmail.com

ABSTRACT

Grover's search algorithm provides a quadratic speedup over classical brute-force methods for unstructured search problems in terms of query complexity and is widely used as a versatile subroutine in numerous quantum algorithms, including those for combinatorial problems with large search spaces. For such problems, it is natural to reduce the effective search space by incorporating problem constraints at the initialization step, which in Grover's algorithm can be achieved by preparing structured initial states that encode constraint information. In this work, we present a systematic framework with a simple preprocessing procedure for constraint-aware initialization in Grover's algorithm, focusing on problems with linear constraints. The proposed framework incorporates cardinality constraints and parity constraints extracted from more general linear constraints. While such structured initial states can reduce the number of oracle queries required to obtain a solution, their preparation incurs additional circuit-level costs. We therefore offer a conservative circuit-level resource analysis, showing that the resulting constraint-aware initialization can improve resource efficiency in terms of gate counts and circuit depth. The proposed framework is further illustrated numerically using the exact-cover problem as a representative application. Overall, our results indicate that this approach serves as a practical baseline for achieving more resource-efficient implementations of Grover's algorithm compared to the standard uniform initialization.

ARTICLE IN PRESS

Introduction

Grover's search algorithm is one of the most prominent quantum algorithms, offering a quadratic speedup over brute-force methods for unstructured search problems[1]. Beyond its original formulation, it has been widely recognized as a versatile component across a broad range of quantum algorithms. In particular, Grover's algorithm underlies amplitude amplification[2,3] and has been adapted for quantum counting[4,5]. It has also been extended from the minimum finding algorithm[6] to more general iterative search schemes for optimization, such as Grover adaptive search[7,8]. Recent work has explored integrating Grover's algorithm with quantum machine learning, for example in robotic kinematic optimization, and has reported potential performance gains on practical high-dimensional instances[9]. This broad applicability highlights the role of Grover's algorithm not only as a breakthrough result in search problems but also as a foundational tool in quantum algorithm design.

Constraint satisfaction problems (CSPs), or more generally, combinatorial optimization problems (COPs), have been actively studied as natural targets for quantum computing approaches, with numerous results reported in recent years[8,10–16]. Since constraints are intrinsic to these problems, they are naturally incorporated into the computational framework. In Grover-based approaches, constraints are typically enforced through constraint-checking operations within the oracle[8,10,11], while variational approaches commonly incorporate them into the problem Hamiltonian through penalty terms[15,17]. Another complementary approach is to encode available constraint information already in the initial state, thereby restricting the search to a smaller or more relevant subspace. Such initialization strategies have been explored in both Grover-based approaches and variational algorithms. For example, several Grover-based studies have investigated problem-tailored initialization for clique search and quadratic assignment problems[18,19], a quantum walk-based reduced initial state for scheduling problems[20], and branching-based iterative state preparation for general linear constraints[21]. In variational settings, related approaches include constraint-preserving mixers and ansätze[13,14,22], as well as tree-based state generation for the knapsack problem[23]. These studies demonstrate that incorporating available constraint information during initialization can effectively reduce the search space before the main computational stage.

In this work, we present a systematic framework for constraint-aware initialization in Grover's algorithm applied to

ARTICLE IN PRESS

combinatorial problems with linear constraints. While previous studies have investigated structured initialization for a variety of combinatorial optimization problems, our framework systematically constructs structured initial states from the underlying linear constraints through a greedy preprocessing procedure. When multiple constraints are available, constructing a structured initial state that simultaneously incorporates them is generally nontrivial because constraints may overlap or require different forms of state preparation. To address this issue, the preprocessing procedure selects a collection of jointly implementable constraint sets, covering cardinality constraints realized by Dicke states[24] as well as parity information extracted from more general linear constraints. We incorporate this parity information via constructions based on Greenberger–Horne–Zeilinger (GHZ) states[25] expressed in the X basis (hereafter referred to as GHZ-type states).

However, implementing constraint-aware initialization incurs additional circuit-level overhead, which motivates a careful resource analysis. We therefore provide a conservative circuit-level resource analysis of the proposed constraint-aware initialization. In this analysis, we focus on representative instances of our framework, namely the Dicke state with Hamming weight one and GHZ-type states, which capture the essential scaling behavior of cardinality and parity constraints. In particular, we examine how the number and size of disjoint constraint sets incorporated into the initial state affect gate counts and circuit depth, and show that the overall resource efficiency improves as additional disjoint sets are included. Even in the worst-case scenario where the preprocessing ultimately retains only a single constraint set, the resulting initialization strategy remains more resource-efficient than the standard uniform initialization. We further illustrate the proposed constraint-aware initialization scheme through numerical simulations on the exact-cover problem as a representative application. The simulation results are consistent with our theoretical analysis, thereby providing numerical support for the proposed framework.

This paper is organized as follows. The Methods section introduces the proposed framework for constraint-aware initialization in Grover's algorithm. The theoretical analysis and numerical simulation results are presented and discussed in the Results and Discussion section. Finally, the Conclusion section summarizes the main findings and outlines possible directions for future work.

Methods

Grover's algorithm with constraint-aware initialization

We consider the problem of identifying an unknown binary string $x \in \{0,1\}^n$, where $n \in \mathbb{N}$, that satisfies a given condition, typically encoded as a Boolean function $f : \{0,1\}^n \to \{0,1\}$. The goal is to find a binary string x such that $f(x) = 1$.

Let $F \subseteq \{0,1\}^n$ denote the search space. In the standard Grover's algorithm, F corresponds to the full search space $\{0,1\}^n$. For the constraint-aware case considered in this work, F denotes a reduced subspace obtained by incorporating some or all of the problem's constraints, such that the subspace F includes the set of all solutions S. The initial state is then prepared as a uniform superposition over all binary strings in F as

$$\begin{aligned} |\psi_F^{(0)}\rangle_{\mathbf{q}_{[n]}} &= V_F|00\cdots0\rangle_{\mathbf{q}_{[n]}} \\ &= \frac{1}{\sqrt{|F|}}\sum_{x\in F}|x\rangle_{\mathbf{q}_{[n]}}, \end{aligned} \tag{1}$$

where $\mathbf{q}_{[n]} = q_1 q_2 \cdots q_n$ and V_F is a unitary operator that generates the desired superposition. For the full search space $F = \{0,1\}^n$, V_F reduces to $H_{q_1} \otimes H_{q_2} \otimes \cdots \otimes H_{q_n}$, where H is the Hadamard operator. We define the solution and non-solution states as

$$|\phi_s\rangle_{\mathbf{q}_{[n]}} = \frac{1}{\sqrt{|S|}}\sum_{x\in S}|x\rangle_{\mathbf{q}_{[n]}} \tag{2}$$

and

$$|\phi_{ns,F}\rangle_{\mathbf{q}_{[n]}} = \frac{1}{\sqrt{|F|-|S|}}\sum_{x\in F\setminus S}|x\rangle_{\mathbf{q}_{[n]}}, \tag{3}$$

respectively. Then, the initial state can be expressed as a linear combination of these two orthonormal components as

$$|\psi_F^{(0)}\rangle_{\mathbf{q}_{[n]}} = \sin\theta|\phi_s\rangle_{\mathbf{q}_{[n]}} + \cos\theta|\phi_{ns,F}\rangle_{\mathbf{q}_{[n]}}, \tag{4}$$

where $\theta = \arcsin(\sqrt{|S|/|F|})$.

The overall procedure of Grover's algorithm consists of preparing the initial state, repeatedly applying the oracle and diffusion operators for an appropriate number of iterations, and finally measuring the state in the computational basis, as illustrated in Figure 1. The oracle operator O_f marks the solutions by flipping their phase:

$$O_f|x\rangle_{\mathbf{q}_{[n]}} = (-1)^{f(x)}|x\rangle_{\mathbf{q}_{[n]}}. \tag{5}$$

ARTICLE IN PRESS

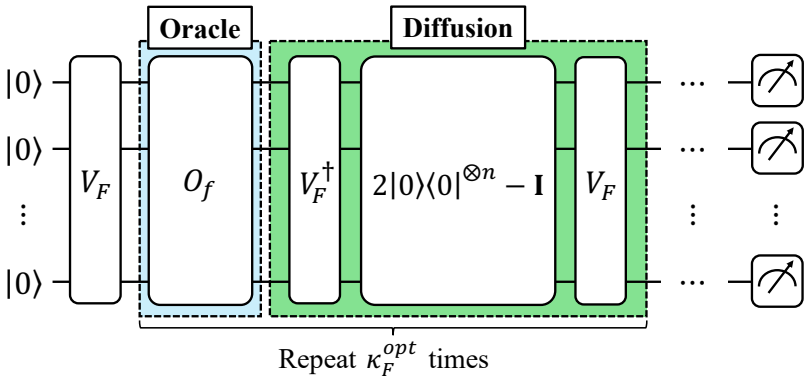


Figure 1. Schematic of Grover's algorithm with constraint-aware initialization. The initial state $|\psi_F^{(0)}\rangle = V_F|0\rangle^{\otimes n}$ is a uniform superposition over the search subspace F, which may encode some or all of the problem's constraints. Each query consists of applying the oracle operator O_f followed by the diffusion operator $R_F = V_F(2|0\rangle\langle 0|^{\otimes n} - \mathbf{I})V_F^\dagger$. After κ_F^{opt} queries, a solution can be obtained with high probability upon measurement.

The diffusion operator is defined as a reflection about the initial state:

$$R_F = 2|\psi_F^{(0)}\rangle\langle\psi_F^{(0)}|_{\mathbf{q}_{[n]}} - \mathbf{I}_{\mathbf{q}_{[n]}}, \tag{6}$$

where $\mathbf{I}$ is the identity operator. We refer to the process of applying the oracle operator followed by the diffusion operator as a query. After performing κ queries starting from the initial state, the quantum state becomes

$$\begin{aligned} |\psi_F^{(\kappa)}\rangle_{\mathbf{q}_{[n]}} &= \left(R_F \cdot O_f\right)^\kappa |\psi_F^{(0)}\rangle_{\mathbf{q}_{[n]}} \\ &= \sin((2\kappa+1)\theta)|\phi_s\rangle_{\mathbf{q}_{[n]}} + \cos((2\kappa+1)\theta)|\phi_{ns,F}\rangle_{\mathbf{q}_{[n]}}. \end{aligned} \tag{7}$$

To maximize the amplitude of the solution state $|\phi_s\rangle$, κ is chosen such that $(2\kappa+1)\theta \approx \pi/2$. As $\theta = \arcsin(\sqrt{|S|/|F|})$, we define the optimal number of queries as

$$\kappa_F^{opt} = \text{round}\left(\frac{\pi}{4\arcsin(\sqrt{|S|/|F|})} - \frac{1}{2}\right). \tag{8}$$

In other words, after κ_F^{opt} queries, measuring the final state in the computational basis yields a solution with high probability. We note that when the number of solutions $|S|$ is unknown, the optimal number of queries can be estimated using techniques such as quantum counting[4,5]. Throughout this work, we assume that $|S|$ is known when evaluating the optimal query number so that the impact of constraint-aware initialization on resource efficiency can be examined independently.

In many practical applications of Grover-based approaches, the oracle is explicitly constructed for the target problem and incorporates problem constraints through constraint-checking operations[8,10,11,26]. In such settings, part of the constraint information can be shifted from the oracle to the initialization stage by preparing an initial state that already satisfies selected constraints. This transfer may increase the cost of state preparation while potentially reducing the complexity of the corresponding oracle construction. The present work investigates this trade-off through constraint-aware initialization and analyzes the resulting circuit-level resource requirements.

Constraint-aware initialization for cardinality constraints

We first consider a class of linear constraints frequently encountered in combinatorial problems, known as cardinality constraints. Such constraints restrict the Hamming weight (the number of ones) of a subset of binary variables and can be written as

$$\sum_{i \in C_j} x_i = b_j, \quad j = 1, \ldots, m, \tag{9}$$

where $C_j \subseteq \{1, \ldots, n\}$ denotes the subset of variable indices associated with the j-th constraint, $x_i \in \{0,1\}$, and $b_j \in \mathbb{N}$. This simple yet important form arises in a variety of combinatorial problems, including the exact-cover problem[15,17,26] and the traveling salesman problem[14,27,28]. While the standard implementation employs a uniform superposition over all 2^n binary

ARTICLE IN PRESS

strings as the initial state, we instead incorporate some of the constraints in Eq. (9) directly into the initial state, so that the search space is reduced to the set of basis states satisfying the selected constraints. We describe how this constraint-aware initialization can be realized using structured quantum states.

To illustrate the idea, let us examine a single cardinality constraint, which requires that the sum of the binary variables in a subset $C \subseteq \{1,\dots,n\}$ equals b. This constraint enforces that exactly b of the variables indexed by C are set to one. A natural way to reflect this constraint in the initial state is to prepare a Dicke state $|D_b^{|C|}\rangle$ on the corresponding qubits, where $|C|$ is the size of the subset C. A Dicke state $|D_\nu^\mu\rangle$ is defined as the equal superposition of all computational basis states of length μ with Hamming weight exactly ν:

$$|D_\nu^\mu\rangle = \frac{1}{\sqrt{\binom{\mu}{\nu}}} \sum_{\substack{x\in\{0,1\}^\mu \\ \mathrm{wt}(x)=\nu}} |x\rangle, \tag{10}$$

where $\mathrm{wt}(x)$ is the Hamming weight of the binary string x. Dicke states can be prepared deterministically without ancillary qubits using the method proposed by Bärtschi and Eidenbenz[29], which requires $O(\mu\nu)$ gates and $O(\mu)$ circuit depth. This indicates that employing Dicke states for constraint-aware initialization can be achieved with reasonable circuit resources.

Given a set of cardinality constraints of the form in Eq. (9), one can in principle construct the initial state by preparing independent Dicke states on disjoint subsets of qubits corresponding to those constraints, while applying Hadamard gates to the remaining qubits. If a collection of disjoint subsets $\{C'_1,\dots,C'_t\}$ can be identified, the resulting initial state can be written, after an appropriate reordering of qubit indices, as

$$|\psi_F^{(0)}\rangle_{\mathbf{q}_{[n]}} = |D_{b'_1}^{|C'_1|}\rangle_{\mathbf{q}_{C'_1}} \otimes \cdots \otimes |D_{b'_t}^{|C'_t|}\rangle_{\mathbf{q}_{C'_t}} \otimes H|0\rangle_{q_{\gamma_t+1}} \otimes \cdots \otimes H|0\rangle_{q_n}, \tag{11}$$

where $\mathbf{q}_\Lambda$ represents the set of qubits associated with the indices in a subset Λ, and $\gamma_t = \sum_{j=1}^t |C'_j|$ is the total number of qubits involved in the preparation of Dicke states. This construction reduces the search space size from 2^n to $2^{n-\gamma_t} \cdot \prod_{j=1}^t \binom{|C'_j|}{b'_j}$. In practice, identifying an optimal collection of disjoint subsets $\{C'_j\}$ that maximizes this reduction is itself a combinatorial problem and can be nontrivial. Nevertheless, a simple greedy procedure in Algorithm 1 can efficiently produce a feasible disjoint subset structure. The constraints are first sorted in ascending order of $\binom{|C_j|}{b_j}/2^{|C_j|}$, so that those imposing the strongest restriction on the feasible search space are prioritized. The algorithm then iteratively selects non-overlapping subsets, corresponding to the case where the overlap threshold is set to $\eta_{\mathrm{th}} = 0$. This preprocessing step requires a classical cost of $O(mk)$, in addition to an initial sorting cost of $O(m\log m)$, where m is the number of constraints and k denotes the typical subset size.

The above construction considers only the case where the selected subsets are mutually disjoint. In many practical settings, however, constraints may share a number of common variables. To handle such cases, the greedy preprocessing can be extended to allow small overlaps among subsets. As described in the second phase of Algorithm 1, subsets whose overlap with the already selected ones is below a chosen threshold η_{th} are sequentially included after removing the shared elements. This additional phase involves only local overlap checks between subsets and thus requires the same classical cost of $O(mk)$ as in the disjoint case. Here, we denote by $\mathscr{C}_{\mathrm{used}}$ the set of variable indices that have already been included in previously selected subsets. For a constraint subset C_j overlapping with $\mathscr{C}_{\mathrm{used}}$ by $r_j = |C_j \cap \mathscr{C}_{\mathrm{used}}| \le \eta_{\mathrm{th}}$, we define the reduced set $R_j = C_j \setminus \mathscr{C}_{\mathrm{used}}$ and relax the corresponding cardinality constraint to

$$\max(0, b_j - r_j) \le \sum_{i\in R_j} x_i \le b_j. \tag{12}$$

This relaxed form defines a valid subspace over the remaining variables without conflicting with the original constraint. The corresponding quantum state is a superposition of Dicke states over the allowed Hamming weights,

$$|D_{b_j,r_j}^{|R_j|}\rangle = \frac{1}{\sqrt{\Omega_j}} \sum_{\nu=b_j^{\min}}^{b_j} \sqrt{\binom{|R_j|}{\nu}} |D_\nu^{|R_j|}\rangle, \tag{13}$$

where $b_j^{\min} = \max(0, b_j - r_j)$ and $\Omega_j = \sum_{\nu=b_j^{\min}}^{b_j} \binom{|R_j|}{\nu}$. Preparing a linear combination of Dicke states of the form $\sum_\nu \alpha_\nu |D_\nu^\mu\rangle$ can be implemented with $O(\mu^2)$ gates and $O(\mu)$ depth[29].

In summary, after preprocessing the constraints and reordering the qubit indices, the initial state can be constructed in the product form

$$|\psi_F^{(0)}\rangle_{\mathbf{q}_{[n]}} = |D_{b'_1}^{|C'_1|}\rangle_{\mathbf{q}_{C'_1}} \otimes \cdots \otimes |D_{b'_t}^{|C'_t|}\rangle_{\mathbf{q}_{C'_t}} \otimes |D_{b'_{t+1},r'_{t+1}}^{|R'_{t+1}|}\rangle_{\mathbf{q}_{R'_{t+1}}} \otimes \cdots \otimes |D_{b'_s,r'_s}^{|R'_s|}\rangle_{\mathbf{q}_{R'_s}} \otimes H|0\rangle_{q_{\gamma_s+1}} \otimes \cdots \otimes H|0\rangle_{q_n}, \tag{14}$$

ARTICLE IN PRESS

Algorithm 1 Classical preprocessing for constraint-aware initialization with cardinality constraints

Input: Collection of pairs $\{(C_j, b_j)\}_{j=1}^{m}$; overlap threshold η_{th}
Output: Selected subset collection $\mathscr{T}$
$\mathscr{T} \leftarrow \emptyset$, $\mathscr{C}_{\text{used}} \leftarrow \emptyset$
$\mathscr{L} \leftarrow \{C_j\}_{j=1}^{m}$ sorted in ascending order of $\binom{|C_j|}{b_j}/2^{|C_j|}$

Phase 1 (disjoint)
for each C_j in $\mathscr{L}$ **do**
  if $C_j \cap \mathscr{C}_{\text{used}} = \emptyset$ **then**
    $\mathscr{T} \leftarrow \mathscr{T} \cup \{C_j\}$
    $\mathscr{C}_{\text{used}} \leftarrow \mathscr{C}_{\text{used}} \cup C_j$
  end if
end for

Phase 2 (overlap)
if $\eta_{\text{th}} > 0$ **then**
  $\mathscr{R} \leftarrow \{C_j \in \mathscr{L} : C_j \notin \mathscr{T}\}$
  for each C_j in $\mathscr{R}$ **do**
    $r \leftarrow |C_j \cap \mathscr{C}_{\text{used}}|$
    if $r \leq \eta_{\text{th}}$ **then**
      $R_j \leftarrow C_j \setminus \mathscr{C}_{\text{used}}$
      if $|R_j| > 0$ **then**
        $\mathscr{T} \leftarrow \mathscr{T} \cup \{R_j\}$
        $\mathscr{C}_{\text{used}} \leftarrow \mathscr{C}_{\text{used}} \cup R_j$
      end if
    end if
  end for
end if
return $\mathscr{T}$

where γ_s is the total number of qubits involved in the preparation of Dicke states and their linear combinations. As a result, the effective search space size is reduced from 2^n to $2^{n-\gamma_s} \cdot \prod_{j=1}^{t} \binom{|C'_j|}{b'_j} \cdot \prod_{j=t+1}^{s} \Omega_j$. Although more general, non-product forms of initialization are conceivable, the product structure in Eq. (14) offers a practical advantage. It provides a systematic and modular framework for embedding multiple constraints while keeping the circuit depth proportional to the largest subset size. Consequently, this approach enables constraint-aware initialization with relatively shallow depth while encoding some of the constraints, and can reduce both the effective search space and the overall circuit-level resource requirements of Grover's algorithm, as will be demonstrated in the Results and Discussion section.

Constraint-aware initialization for parity constraints extracted from linear constraints

We next examine a more general linear constraint of the form

$$\sum_{i \in C} \alpha_i x_i = b, \tag{15}$$

where $C \subseteq \{1, \ldots, n\}$, $x_i \in \{0, 1\}$, and $\alpha_i, b \in \mathbb{Z}$. Without loss of generality, we assume that the greatest common divisor of all coefficients $\{\alpha_i\}_{i \in C}$ and b is one. In contrast to the cardinality constraint discussed in the previous subsection, the number of ones among $\{x_i\}_{i \in C}$ cannot be predetermined in this case. Thus, the method based on the Dicke state is not applicable. To construct a feasible initialization in this setting, we instead exploit the parity structure of the coefficients. Let $\hat{C} = \{i \in C | \alpha_i \text{ is odd}\}$. By reducing Eq. (15) modulo 2, we obtain

$$\sum_{i \in \hat{C}} x_i \equiv b \pmod 2, \tag{16}$$

ARTICLE IN PRESS

Algorithm 2 Classical preprocessing for constraint-aware initialization using parity constraints extracted from linear constraints

Input: Collection of triples $\{(C_j, \{\alpha_{j,i}\}_{i\in C_j}, b_j)\}_{j=1}^{m}$
Output: Selected subset collection $\mathscr{T}$
$\mathscr{T} \leftarrow \emptyset$, $\mathscr{C}_{\text{used}} \leftarrow \emptyset$
for $j = 1, \ldots, m$ **do**
    $\hat{C}_j \leftarrow \{\, i \in C_j : \alpha_{j,i} \text{ is odd} \,\}$
end for
$\mathscr{L} \leftarrow \{\hat{C}_j\}_{j=1}^{m}$ sorted in ascending order of $|\hat{C}_j|$
(Disjoint subset selection)
for each $\hat{C}_j$ in $\mathscr{L}$ **do**
    if $\hat{C}_j \cap \mathscr{C}_{\text{used}} = \emptyset$ **then**
        $\mathscr{T} \leftarrow \mathscr{T} \cup \{\hat{C}_j\}$
        $\mathscr{C}_{\text{used}} \leftarrow \mathscr{C}_{\text{used}} \cup \hat{C}_j$
    end if
end for
return $\mathscr{T}$

which implies that if b is even (odd), then the number of ones among $\{x_i\}_{i\in\hat{C}}$ must also be even (odd). This parity condition can be encoded using a GHZ-type entangled state of the form

$$\begin{aligned} |GHZ^{(X)}_{\mu,\nu}\rangle &= \frac{1}{\sqrt{2}}\left(|+\rangle^{\otimes\mu} + (-1)^{\nu}|-\rangle^{\otimes\mu}\right) \\ &= \frac{1}{\sqrt{2^{\mu-1}}} \sum_{\substack{x\in\{0,1\}^{\mu} \\ \mathrm{wt}(x)\equiv\nu \pmod 2}} |x\rangle, \end{aligned} \tag{17}$$

where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$, $\mu \in \mathbb{N}$, and $\nu \in \{0,1\}$. In the present setting, we set $\mu = |\hat{C}|$ and take $\nu = b \pmod 2$, reflecting the parity condition in Eq. (16).

Now suppose that a set of linear constraints of the form

$$\sum_{i\in C_j} \alpha_{j,i} x_i = b_j, \quad j = 1, \ldots, m, \tag{18}$$

is given, where, for each j, the coefficients $\{\alpha_{j,i}\}$ and b_j have greatest common divisor one. Each constraint induces a parity condition on the subset $\hat{C}_j = \{i \in C_j | \alpha_{j,i} \text{ is odd}\}$. If one can identify disjoint subsets $\{\hat{C}'_1, \ldots, \hat{C}'_t\}$, then, after reordering the qubit indices, the initial state can be written as

$$|\psi_F^{(0)}\rangle_{\mathbf{q}_{[n]}} = |GHZ^{(X)}_{|\hat{C}'_1|,b'_1}\rangle_{\mathbf{q}_{\hat{C}'_1}} \otimes \cdots \otimes |GHZ^{(X)}_{|\hat{C}'_t|,b'_t}\rangle_{\mathbf{q}_{\hat{C}'_t}} \otimes H|0\rangle_{q_{\hat{\gamma}_t+1}} \otimes \cdots \otimes H|0\rangle_{q_n}, \tag{19}$$

where $\hat{\gamma}_t = \sum_{j=1}^{t} |\hat{C}'_j|$ denotes the total number of qubits involved in the parity encoding. Since each constraint effectively reduces the search space by half, the resulting search space size becomes 2^{n-t}, independent of the subset sizes $|\hat{C}_j|$. To maximize the overall reduction, it is therefore desirable to identify as many disjoint subsets as possible. To this end, we employ a simple greedy preprocessing procedure that iteratively selects disjoint parity subsets in ascending order of their sizes, as described in Algorithm 2.

When both cardinality and parity constraints are present, the initialization can be systematically constructed by combining Dicke states and GHZ-type states. Since encoding a cardinality constraint on a subset C_j via Dicke state preparation reduces the search space by a factor of $\binom{|C_j|}{b_j}/2^{|C_j|}$, whereas each parity constraint reduces it by exactly one half under the proposed initialization scheme, it can be advantageous in terms of search space reduction to prioritize subsets associated with cardinality constraints before applying the parity-based selection to the remaining constraints. This ordering enables an efficient reduction of the overall search space within a systematic framework.

Results and Discussion

Resource requirement analysis

We evaluate the overall resource requirements associated with the constraint-aware initialization considered in this work. In particular, we examine how structural features of the imposed constraints, such as the number of constraints incorporated into

ARTICLE IN PRESS

the initialization and the size of the constraint sets, affect the resulting resource requirements. The resource metrics considered here include the total gate count, the circuit depth, and the number of two-qubit gates. The total gate count serves as a measure of circuit complexity, the circuit depth reflects the computation time, and the number of two-qubit gates provides a practical indicator of error sensitivity in experimental implementations.

Throughout this resource analysis, we focus on two representative cases, namely the states $|D_1^{\mu}\rangle$ and $|GHZ_{\mu,\nu}^{(X)}\rangle$, which encode cardinality and parity constraints, respectively. Although our quantitative analysis centers on these representative examples, the underlying arguments rely only on generic features such as the reduction of the effective search space and the overhead associated with state preparation. As a result, while the explicit form of the efficiency condition may depend on the specific structure of the initial state preparation, we expect that similar criteria and analytical frameworks can be employed to assess the relative efficiency of more general forms of constraint-aware initialization.

For an initialization strategy σ, let $\mathscr{S}_\sigma$ denote the cost of preparing the initial state, and $\mathscr{O}_\sigma$ the cost of implementing the oracle. Each strategy σ induces a search space F_σ, and the corresponding optimal number of queries $\kappa_{F_\sigma}^{opt}$ defined in Eq. (8) is abbreviated as κ_σ. Let $\mathscr{D}$ denote the cost of implementing the unitary $2|0\rangle\langle 0|^{\otimes n} - \mathbf{I}$, which appears in the diffusion operator and is independent of the choice of initialization strategy. Assuming that implementing a unitary and its inverse incurs the same cost, the total resource requirement can be expressed as

$$\mathscr{R}_\sigma = \mathscr{S}_\sigma + (\mathscr{O}_\sigma + \mathscr{D} + 2\mathscr{S}_\sigma)\,\kappa_\sigma, \tag{20}$$

where the first term corresponds to the one-time cost of state preparation, while the second term accounts for repeated applications of the oracle and both the forward and inverse state preparation operators in each query. In practice, the overall cost may be further reduced through optimizations such as commuting operations. However, adopting the expression in Eq. (20) provides a consistent and transparent basis for comparing the relative efficiency of different initialization strategies. We say that strategy τ is more efficient than strategy σ (with respect to a given metric) if $\mathscr{R}_\tau < \mathscr{R}_\sigma$.

Before analyzing specific initialization schemes, we derive a sufficient condition under which one initialization strategy is more resource-efficient than another.

Lemma 1 *For strategies σ and τ, suppose that $\mathscr{O}_\tau \le \mathscr{O}_\sigma$ and $\kappa_\tau < \kappa_\sigma$. If*

$$\mathscr{O}_\sigma + \mathscr{D} > \left(\frac{2\kappa_\tau + 1}{\kappa_\sigma - \kappa_\tau}\right)(\mathscr{S}_\tau - \mathscr{S}_\sigma) - 2\mathscr{S}_\sigma, \tag{21}$$

then $\mathscr{R}_\tau < \mathscr{R}_\sigma$.

Proof. Since $\mathscr{O}_\tau \le \mathscr{O}_\sigma$,

$$\mathscr{R}_\tau = \mathscr{S}_\tau + (\mathscr{O}_\tau + \mathscr{D} + 2\mathscr{S}_\tau)\,\kappa_\tau \le \mathscr{S}_\tau + (\mathscr{O}_\sigma + \mathscr{D} + 2\mathscr{S}_\tau)\,\kappa_\tau. \tag{22}$$

Hence, whenever

$$\mathscr{S}_\tau + (\mathscr{O}_\sigma + \mathscr{D} + 2\mathscr{S}_\tau)\,\kappa_\tau < \mathscr{R}_\sigma, \tag{23}$$

it follows that $\mathscr{R}_\tau < \mathscr{R}_\sigma$. Substituting $\mathscr{R}_\sigma = \mathscr{S}_\sigma + (\mathscr{O}_\sigma + \mathscr{D} + 2\mathscr{S}_\sigma)\,\kappa_\sigma$ into Eq. (23) and rearranging gives

$$\begin{aligned}(\mathscr{O}_\sigma + \mathscr{D})(\kappa_\sigma - \kappa_\tau) &> 2\mathscr{S}_\tau\kappa_\tau - 2\mathscr{S}_\sigma\kappa_\sigma + \mathscr{S}_\tau - \mathscr{S}_\sigma \\ &= 2\kappa_\tau(\mathscr{S}_\tau - \mathscr{S}_\sigma) - 2\mathscr{S}_\sigma(\kappa_\sigma - \kappa_\tau) + \mathscr{S}_\tau - \mathscr{S}_\sigma.\end{aligned} \tag{24}$$

Since $\kappa_\tau < \kappa_\sigma$, dividing both sides by $\kappa_\sigma - \kappa_\tau$ yields the criterion in Eq. (21). □

Cardinality constraints

Let us first consider the case of cardinality constraints and analyze how the incremental incorporation of disjoint constraint sets influences the overall resource requirement. To this end, we introduce a sequence of strategies σ_i, where σ_0 corresponds to the standard uniform initialization without constraints, and σ_i denotes the strategy obtained by incorporating i disjoint constraint sets into the initial state preparation. Each successive strategy σ_{i+1} is constructed by adding one additional disjoint constraint set of size μ_{i+1} to σ_i, and we derive conditions under which σ_{i+1} yields a lower resource cost than σ_i. When the additional constraint set is incorporated into the initial state, the corresponding constraint no longer needs to be enforced within the oracle. Hence, the oracle used for σ_i can be reused for σ_{i+1}, which implies $\mathscr{O}_{\sigma_{i+1}} \le \mathscr{O}_{\sigma_i}$. Moreover, if the oracle is constructed by sequentially applying constraint-checking operations for each constraint, then incorporating the additional constraint into the initialization removes the corresponding checking operation from the oracle and may therefore lead to a strict reduction in oracle cost. The following proposition formalizes the comparison for the case in which the newly added constraint set is prepared as a Dicke state $|D_1^{\mu_{i+1}}\rangle$.

ARTICLE IN PRESS

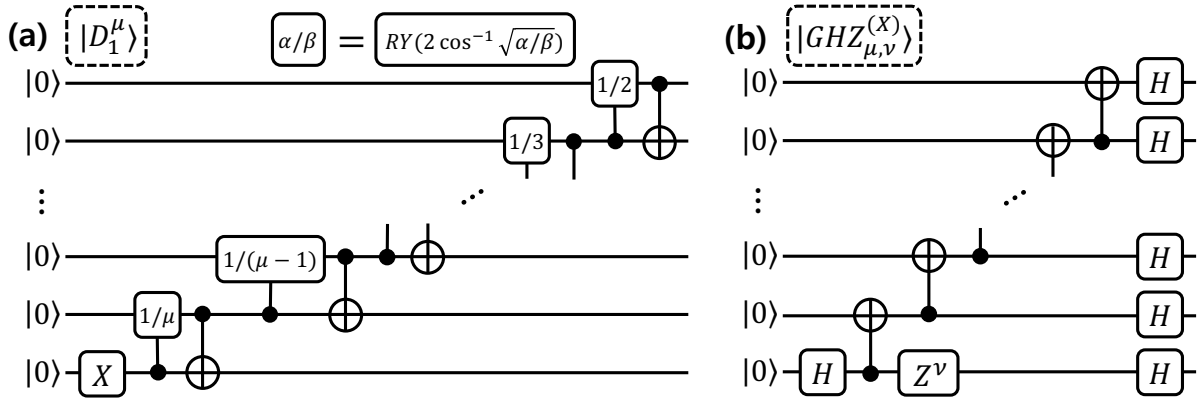


Figure 2. (a) Quantum circuit for preparing $|D_1^\mu\rangle$ on μ qubits. (b) Quantum circuit for preparing $|GHZ_{\mu,\nu}^{(X)}\rangle$ on μ qubits. From the circuits, the total gate count, the two-qubit gate count, and the circuit depth are $2\mu-1$, $2\mu-2$, and $2\mu-1$ for (a), and $2\mu+1$, $\mu-1$, and $\mu+1$ for (b), respectively.

Proposition 2 *Suppose that strategy σ_{i+1} is obtained from σ_i by incorporating one additional disjoint constraint set represented by $|D_1^\mu\rangle$, where we simply denote μ_{i+1} by $\mu \geq 2$. Suppose further that $\mathscr{O}_{\sigma_{i+1}} \leq \mathscr{O}_{\sigma_i}$, $|F_{\sigma_i}| \geq 64|S|$, and $|F_{\sigma_{i+1}}| \geq 2|S|$, where $|S| > 0$ is the number of solutions. If*

$$\mathscr{O}_{\sigma_i} + \mathscr{D} > \left(\frac{24\pi}{21\sqrt{\frac{2^\mu}{\mu}} - 8\pi}\right)\left(\mathscr{S}_{\sigma_{i+1}} - \mathscr{S}_{\sigma_i}\right) - 2\mathscr{S}_{\sigma_i}, \tag{25}$$

then strategy σ_{i+1} is more efficient than strategy σ_i. In particular, when the Dicke state $|D_1^\mu\rangle$ is prepared using the circuit shown in Figure 2(a), the sufficient condition simplifies to

$$\mathscr{O}_{\sigma_i} + \mathscr{D} \geq 67 \tag{26}$$

for each of the considered resource metrics (total gate count, circuit depth, and number of two-qubit gates).

Proof. Let $x_i = \sqrt{|S|/|F_{\sigma_i}|}$ and $x_{i+1} = \sqrt{|S|/|F_{\sigma_{i+1}}|}$. Then $x_{i+1} \geq \sqrt{2}x_i$ for $\mu \geq 2$ because $|F_{\sigma_i}|/|F_{\sigma_{i+1}}| = 2^\mu/\mu$. Since $\arcsin x$ is strictly convex on $(0,1)$ with $\arcsin 0 = 0$ and $0 < x_i \leq 1/8$, it follows that

$$\frac{1}{\arcsin x_{i+1}} < \frac{1}{\sqrt{2}\arcsin x_i}. \tag{27}$$

Using $\kappa_{\sigma_i} \geq \pi/(4\arcsin x_i) - 1$ and $\kappa_{\sigma_{i+1}} \leq \pi/(4\arcsin x_{i+1})$, we obtain

$$\kappa_{\sigma_i} - \kappa_{\sigma_{i+1}} > \frac{\pi}{4\arcsin x_i}\left(1 - \frac{1}{\sqrt{2}}\right) - 1. \tag{28}$$

Since $|F_{\sigma_i}| \geq 64|S|$ and $\arcsin x$ is increasing on $[0,1]$, the right-hand side is positive, which implies $\kappa_{\sigma_i} > \kappa_{\sigma_{i+1}}$. Together with the assumption $\mathscr{O}_{\sigma_{i+1}} \leq \mathscr{O}_{\sigma_i}$, Lemma 1 can be applied and it remains to bound

$$\frac{2\kappa_{\sigma_{i+1}} + 1}{\kappa_{\sigma_i} - \kappa_{\sigma_{i+1}}}. \tag{29}$$

Since $x_i \leq 1/8 < 0.17$, it follows from the bound $\arcsin x \leq 8\pi x/25$ for $0 \leq x \leq 0.17$ that

$$\kappa_{\sigma_i} \geq \frac{\pi}{4\arcsin x_i} - 1 \geq \frac{21}{32x_i}. \tag{30}$$

On the other hand, since $x \leq \arcsin x$ for $0 \leq x \leq 1$,

$$\kappa_{\sigma_{i+1}} \leq \frac{\pi}{4\arcsin x_{i+1}} \leq \frac{\pi}{4x_{i+1}}. \tag{31}$$

ARTICLE IN PRESS

Therefore, using $x_{i+1}/x_i = \sqrt{2^\mu/\mu}$ and the fact that $|F_{\sigma_{i+1}}| \geq 2|S|$ implies $1 \leq \pi/(4x_{i+1})$,

$$\frac{2\kappa_{\sigma_{i+1}}+1}{\kappa_{\sigma_i}-\kappa_{\sigma_{i+1}}} \leq \frac{\frac{3\pi}{x_i}\sqrt{\frac{\mu}{2^\mu}}}{4\kappa_{\sigma_i}-\frac{\pi}{x_i}\sqrt{\frac{\mu}{2^\mu}}} \leq \frac{24\pi}{21\sqrt{\frac{2^\mu}{\mu}}-8\pi}, \tag{32}$$

which gives the criterion in Eq. (25).

Under the specific implementation where the Dicke state $|D_1^\mu\rangle$ is prepared using the circuit shown in Figure 2(a), the additional cost incurred by incorporating one more constraint set satisfies

$$\mathscr{S}_{\sigma_{i+1}} - \mathscr{S}_{\sigma_i} \leq 2\mu, \tag{33}$$

for each of the considered resource metrics, such as the total gate count, the circuit depth, and the number of two-qubit gates. Therefore, we obtain a sufficient condition for this specific implementation,

$$\mathscr{O}_{\sigma_i} + \mathscr{D} > \frac{48\pi\mu}{21\sqrt{\frac{2^\mu}{\mu}}-8\pi}, \tag{34}$$

where we omit the negative term $-2\mathscr{S}_{\sigma_i}$. It can be shown that the right-hand side decreases monotonically for $\mu \geq 2$, and its value at $\mu = 2$ is approximately 66.1. Hence, in this case, the inequality in Eq. (26) guarantees that strategy σ_{i+1} is more efficient than strategy σ_i. □

An interesting observation is that the cost $\mathscr{D}$ of implementing $2|0\rangle\langle 0|^{\otimes n} - \mathbf{I}$ alone is typically sufficient to satisfy the inequality in Eq. (26), even without taking the oracle construction into account. In practice, $\mathscr{D}$ is dominated by the implementation of the multi-controlled Z gate (which can be expressed using Hadamard gates and a multi-controlled X gate). Decomposing this component using a standard ancilla-assisted construction[30] yields gate counts and circuit depths that readily exceed the threshold value of 67 even for a relatively small number of qubits. The same conclusion holds for ancilla-free decompositions, which generally require substantially higher resource costs[31]. Consequently, in many practical settings, the sufficient condition in Eq. (26) can be satisfied even before accounting for the oracle construction cost. This suggests that incorporating additional disjoint cardinality constraints into the initialization can improve overall resource efficiency. Notably, even when only a single constraint set is used, the resulting initialization strategy can outperform the standard approach, highlighting that effectively exploiting readily available constraints can be practically meaningful without exhaustive constraint coverage.

We next study how the size of an additional constraint set affects resource efficiency with $|D_1^\mu\rangle$. Unlike the preceding comparison, which compares a strategy that incorporates an additional constraint with one that does not, the comparison considered here concerns two strategies that both incorporate an additional constraint but differ in the size of the additional constraint. Consequently, the relation between the corresponding oracle costs is more implementation-dependent, and the oracle-reuse argument used in Proposition 2 and 4 does not directly apply. For example, for a fixed cardinality value b, checking whether a set of variables satisfies $\sum_{i\in C} x_i = b$ may require a more costly circuit as the size of C increases. In this case, encoding the constraint into the initialization removes the corresponding constraint-checking operation from the oracle, so eliminating the checking operation for a larger cardinality set may result in a lower oracle implementation cost. The following proposition formalizes one such setting.

Proposition 3 *Suppose that two strategies σ and σ' both incorporate an additional disjoint constraint set into the initialization and differ in the size of the additional constraint set, which is represented by $|D_1^\mu\rangle$ for σ and $|D_1^{\mu+1}\rangle$ for σ', where $\mu \geq 2$. Suppose further that $\mathscr{O}_{\sigma'} \leq \mathscr{O}_\sigma$, $|F_\sigma| \geq 100|S|$, and $|F_{\sigma'}| \geq 2|S|$, where $|S| > 0$ is the number of solutions. If*

$$\mathscr{O}_\sigma + \mathscr{D} > \left(\frac{120\pi}{109\sqrt{\frac{2\mu}{\mu+1}}-40\pi}\right)(\mathscr{S}_{\sigma'} - \mathscr{S}_\sigma) - 2\mathscr{S}_\sigma, \tag{35}$$

then strategy σ' is more efficient than strategy σ. In particular, when the Dicke state $|D_1^\mu\rangle$ is prepared using the circuit shown in Figure 2(a), the sufficient condition simplifies to

$$\mathscr{O}_\sigma + \mathscr{D} \geq 97, \tag{36}$$

for $\mu \geq 3$. This condition applies to each of the considered resource metrics (total gate count, circuit depth, and number of two-qubit gates).

ARTICLE IN PRESS

Proof. Let $x = \sqrt{|S|/|F_\sigma|}$ and $x' = \sqrt{|S|/|F_{\sigma'}|}$. Then $x' \geq \sqrt{4/3}x$ for $\mu \geq 2$ because $|F_\sigma|/|F_{\sigma'}| = 2\mu/(\mu+1)$. Repeating the argument used in Proposition 2 under the conditions $\mathscr{O}_{\sigma'} \leq \mathscr{O}_\sigma$, $|F_\sigma| \geq 100|S|$, and $|F_{\sigma'}| \geq 2|S|$, one obtains $\kappa_{\sigma'} < \kappa_\sigma$ and

$$\frac{2\kappa_{\sigma'}+1}{\kappa_\sigma - \kappa_{\sigma'}} \leq \frac{120\pi}{109\sqrt{\frac{2\mu}{\mu+1}} - 40\pi}. \tag{37}$$

Applying Lemma 1 then yields the criterion in Eq. (35).

Under the specific implementation where the Dicke state $|D_1^\mu\rangle$ is prepared using the circuit shown in Figure 2(a), we have

$$\mathscr{S}_{\sigma'} - \mathscr{S}_\sigma = 2. \tag{38}$$

Substituting this relation into Eq. (35) gives

$$\mathscr{O}_\sigma + \mathscr{D} > \frac{240\pi}{109\sqrt{\frac{2\mu}{\mu+1}} - 40\pi}. \tag{39}$$

The right-hand side is decreasing for $\mu \geq 2$, and its value drops sharply when μ increases from 2 to 3, after which the decrease becomes more gradual. Evaluating the bound at $\mu = 3$ provides a simple and conservative sufficient condition that holds for all $\mu \geq 3$, yielding the inequality in Eq. (36). □

Proposition 3, established for the $|D_1^\mu\rangle$-based initialization, shows that incorporating a larger constraint set can improve resource efficiency under the considered oracle-cost condition. This observation is consistent with the preprocessing in Algorithm 1, which seeks to combine constraints into larger sets whenever possible. Although the preprocessing strategy may not be optimal, it remains meaningful as a simple baseline for exploiting the potential advantages of larger constraint sets in a conservative setting.

The above results are established for the weight-one Dicke state $|D_1^\mu\rangle$. For a general weight-b Dicke state $|D_b^\mu\rangle$, an analogous comparison holds, but the constants appearing in the criteria depend on the search space ratios $|F_{\sigma_i}|/|F_{\sigma_{i+1}}|$ and $|F_\sigma|/|F_{\sigma'}|$, as well as the ratio $|S|/|F|$ of the number of solutions to the search space size. A detailed analysis of the threshold behavior for general b, including explicit constants and circuit-level decompositions, is left for further investigation. Furthermore, identifying effective initialization strategies across constraint sets of varying cardinality could lead to a more efficient preprocessing procedure.

Parity constraints extracted from linear constraints

In analogy to the discussion of cardinality constraints in the previous subsection, we examine how parity constraints affect the resource cost of the initialization. A key difference is that a parity constraint implemented via a GHZ-type state $|GHZ_{\mu,\nu}^{(X)}\rangle$ always reduces the search space by half, independent of the constraint set size μ. As a result, increasing μ does not yield any further reduction of the search space, but instead incurs a higher circuit cost. It therefore follows that, for parity constraints, applying constraints on smaller subsets of qubits is generally more efficient in terms of overall resource cost.

We now examine how the number of imposed parity constraints affects the overall resource requirements. As before, we consider a sequence of strategies σ_i, where σ_0 denotes the standard initialization and each successive strategy incorporates one additional disjoint constraint set of size μ. Adding such a set via $|GHZ_{\mu,\nu}^{(X)}\rangle$ halves the search space, that is, $|F_{\sigma_i}|/|F_{\sigma_{i+1}}| = 2$. As in the case of cardinality constraints, we consider oracle constructions in which incorporating an additional parity constraint into the initialization does not increase the oracle cost. Meanwhile, when $|GHZ_{\mu,\nu}^{(X)}\rangle$ is implemented using the circuit in Figure 2(b), then we have $\mathscr{S}_{\sigma_{i+1}} - \mathscr{S}_{\sigma_i} \leq 2\mu + 1$. Hence, by applying the same comparison argument as in Proposition 2, we obtain the following result.

Proposition 4 *Suppose that strategy σ_{i+1} is obtained from σ_i by incorporating one additional disjoint constraint set represented by $|GHZ_{\mu,\nu}^{(X)}\rangle$, where $\mu \geq 2$. Suppose further that $\mathscr{O}_{\sigma_{i+1}} \leq \mathscr{O}_{\sigma_i}$, $|F_{\sigma_i}| \geq 64|S|$, and $|F_{\sigma_{i+1}}| \geq 2|S|$, where $|S| > 0$ is the number of solutions. If*

$$\mathscr{O}_{\sigma_i} + \mathscr{D} > \left(\frac{24\pi}{21\sqrt{2} - 8\pi}\right)\left(\mathscr{S}_{\sigma_{i+1}} - \mathscr{S}_{\sigma_i}\right) - 2\mathscr{S}_{\sigma_i}, \tag{40}$$

then strategy σ_{i+1} is more efficient than strategy σ_i. When preparing $|GHZ_{\mu,\nu}^{(X)}\rangle$ using the circuit in Figure 2(b), we have a simplified sufficient condition:

$$\mathscr{O}_{\sigma_i} + \mathscr{D} > \frac{24\pi(2\mu+1)}{21\sqrt{2} - 8\pi} \tag{41}$$

for each of the considered resource metrics (total gate count, circuit depth, and number of two-qubit gates).

ARTICLE IN PRESS

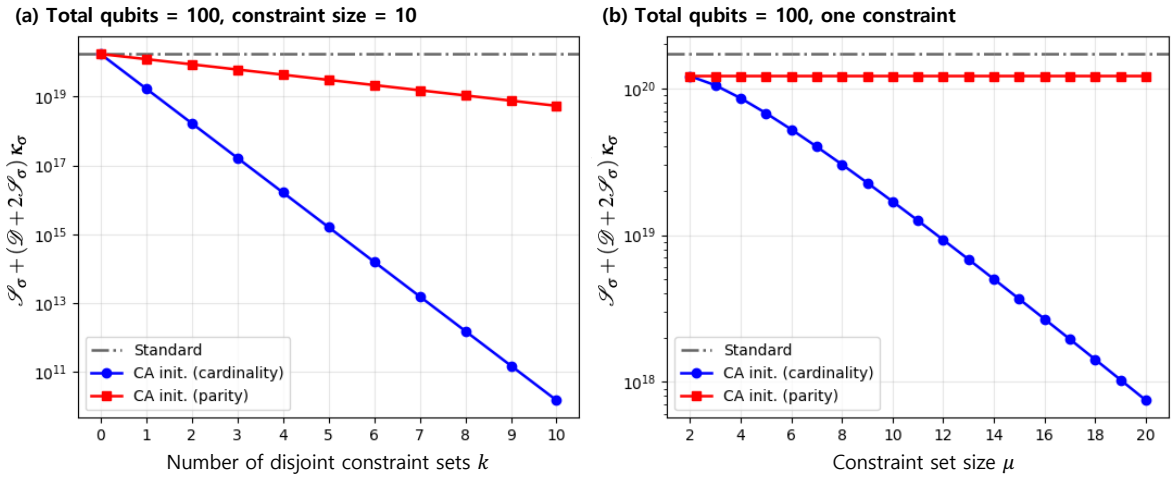


Figure 3. Resource cost of constraint-aware initialization for $n = 100$ qubits with $|S| = 1$. The vertical axis shows the non-oracle resource cost $\mathscr{S}_\sigma + (\mathscr{D} + 2\mathscr{S}_\sigma)\kappa_\sigma$ measured in total gate count, where the multi-controlled X gate in $\mathscr{D}$ is decomposed using the gate set $\{RZ, H, CNOT\}$. The dash-dot line (Standard) corresponds to the uniform superposition initialization $|+\rangle^{\otimes 100}$. (a) Resource cost as the number of incorporated disjoint constraint sets k increases, with each constraint set acting on 10 qubits. Cardinality constraints are implemented using $|D_1^{10}\rangle$, whereas parity constraints are implemented using $|GHZ_{10,1}^{(X)}\rangle$. Both approaches reduce the resource cost relative to the standard baseline, with cardinality constraints achieving substantially greater reductions than parity constraints. (b) Resource cost as the size μ of a single incorporated constraint set increases. Cardinality and parity constraints are implemented using $|D_1^{\mu}\rangle$ and $|GHZ_{\mu,1}^{(X)}\rangle$, respectively. The resource cost decreases rapidly with μ for cardinality constraints, while it increases slightly yet remains below the standard baseline for parity constraints.

The right-hand side of the inequality in Eq. (41) scales linearly with the constraint set size μ. Even so, when μ is small compared to the total number of qubits, the inequality can often be satisfied by accounting only for the cost $\mathscr{D}$. For larger values of μ, the condition can still be met in practice once the oracle construction cost $\mathscr{O}$ is taken into account. Therefore, incorporating disjoint constraint sets, even with relatively large μ, can still be advantageous in terms of overall resource efficiency under the considered oracle-cost condition.

To numerically illustrate the analytical resource scaling of the proposed framework, we compute the non-oracle resource cost

$$\mathscr{S}_\sigma + (\mathscr{D} + 2\mathscr{S}_\sigma)\kappa_\sigma. \tag{42}$$

Figure 3 shows that the non-oracle resource cost decreases independently of the oracle implementation. Together with the reduced optimal query number and the oracle-cost relations established in Propositions 2-4, this implies a reduction in the overall resource cost when constraint-aware initialization is employed. To assess the scale of the diffusion cost, we first examine $\mathscr{D}$, which corresponds to implementing the unitary $2|0\rangle\langle 0|^{\otimes n} - \mathbf{I}$. The operator $2|0\rangle\langle 0|^{\otimes n} - \mathbf{I}$ is implemented using the Hadamard gate H, Pauli-X and Pauli-Z gates, together with a multi-controlled X gate[30,32]. Here, we further decompose the multi-controlled X gate into one- and two-qubit gates over the gate set $\{RZ, H, CNOT\}$, which yielded a relatively compact one- and two-qubit gate decomposition among the implementations we examined using IBM's Qiskit 2.1.1. For example, even for $n = 10$, the resulting circuit requires 713 one-qubit gates and 444 two-qubit gates with circuit depth of 825, and the cost increases rapidly as n grows. This already exceeds the thresholds appearing in Eqs. (26) and (36), indicating that the sufficient conditions of the propositions can be satisfied in practice without requiring an exceptionally costly oracle.

Figure 3 shows the resulting non-oracle resource cost measured in total gate count for representative cardinality- and parity-based initializations with $n = 100$ and $|S| = 1$. The resource counts used in this analysis are summarized in Table 1, and the optimal number of queries is computed according to Eq. (8). In Figure 3(a), the size of each incorporated constraint set is fixed to 10 qubits, and the number of disjoint constraint sets is varied. For cardinality and parity constraints, the initial states are prepared as $|D_1^{10}\rangle^{\otimes k} \otimes |+\rangle^{\otimes 100-10k}$ and $|GHZ_{10,1}^{(X)}\rangle^{\otimes k} \otimes |+\rangle^{\otimes 100-10k}$. Both cardinality and parity constraints reduce the resource cost relative to the standard uniform initialization $|+\rangle^{100}$, with the reduction being substantially greater for cardinality constraints. Moreover, the resource cost decreases as additional disjoint constraint sets are incorporated, consistent with the trends predicted by Propositions 2 and 4. In Figure 3(b), a single constraint set is incorporated and its size μ is varied. The

ARTICLE IN PRESS

Component	One-qubit	Two-qubit	Total	Depth
$\|+\rangle^{\otimes 100}$	100	-	100	1
$\|D_1^\mu\rangle^{\otimes k} \otimes \|+\rangle^{\otimes 100-\mu k}$	$100-(\mu-1)k$	$2(\mu-1)k$	$100+(\mu-1)k$	$2\mu-1$
$\|GHZ_{\mu,1}^{(X)}\rangle^{\otimes k} \otimes \|+\rangle^{\otimes 100-\mu k}$	$100+2k$	$(\mu-1)k$	$100+(\mu+1)k$	$\mu+1$
$2\|0\rangle\langle 0\|^{\otimes 100}-\mathbf{I}$	116813	76764	193577	186045

Table 1. Resource counts for representative initialization circuits and the diffusion operator. The cost of implementing $2|0\rangle\langle 0|^{\otimes 100}-\mathbf{I}$ is estimated after decomposing the multi-controlled X gate into the gate set $\{RZ, H, CNOT\}$, whereas the initialization cost follows the circuit construction in Figure 2, where each controlled rotation is counted as a single two-qubit gate.

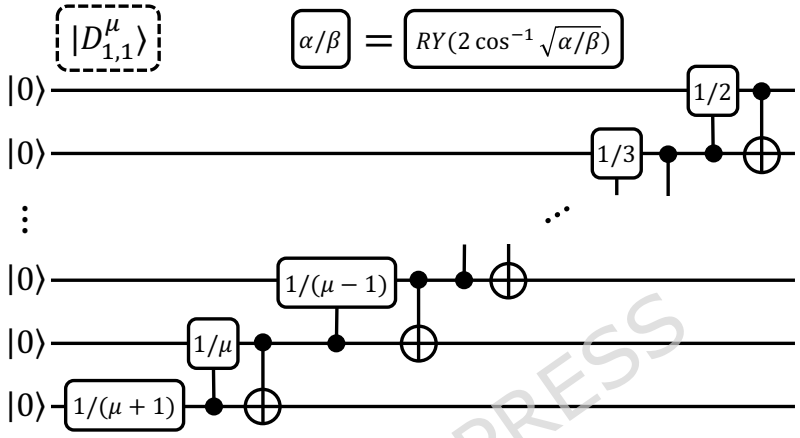


Figure 4. Quantum circuit for preparing $|D_{1,1}^{\mu}\rangle$ on μ qubits.

resource cost decreases rapidly for cardinality constraints, as predicted by Proposition 3, whereas it slightly increases for parity constraints because the search-space reduction remains fixed at a factor of two while the state-preparation cost grows with μ. Nevertheless, the parity-based initialization remains more resource-efficient than the standard uniform initialization throughout the range considered, providing additional numerical support for the resource-efficiency trends predicted by Propositions 2-4.

Simulation results for cardinality constraints

To illustrate the effectiveness of the proposed initialization scheme, we consider the exact-cover problem as a representative combinatorial problem. Given a collection $\mathscr{A} = \{A_1, A_2, \ldots, A_n\}$ of subsets of a universal set $U = \{u_1, u_2, \ldots, u_m\}$, a subcollection $\mathscr{A}' \subseteq \mathscr{A}$ is said to form an exact-cover of U if it satisfies the following two conditions:

(i) Any two distinct subsets $A_i, A_j \in \mathscr{A}'$ are disjoint:

$$A_i \cap A_j = \emptyset.$$

(ii) The union of all subsets in $\mathscr{A}'$ covers the entire set U:

$$\bigcup_{A_i \in \mathscr{A}'} A_i = U.$$

The objective is to find such a subcollection $\mathscr{A}'$ among the 2^n possible choices.

Equivalently, the problem can be formulated in terms of a binary string $x = x_1 x_2 \ldots x_n \in \{0,1\}^n$, where $x_i = 1$ indicates that subset A_i is selected. For each $j \in \{1, 2, ..., m\}$, we define

$$C_j = \{i \,|\, u_j \in A_i\}, \tag{43}$$

the set of indices of subsets that contain u_j. The exact-cover condition then requires that

$$\sum_{i \in C_j} x_i = 1, \quad j = 1, \ldots, m, \tag{44}$$

ARTICLE IN PRESS

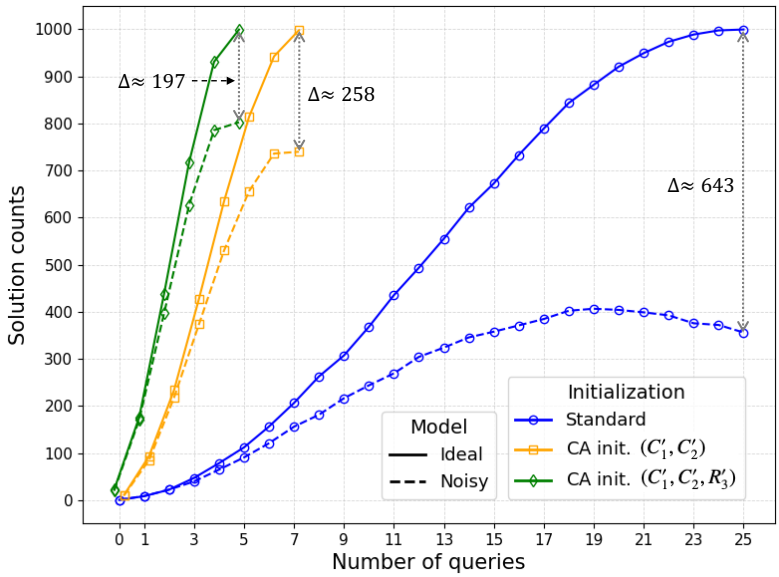


Figure 5. Comparison of the solution counts for the exact-cover instance in Eq. (46) under different initialization strategies. The standard uniform initialization is compared with two constraint-aware initialization (CA init.) schemes incorporating (C_1', C_2') and (C_1', C_2', R_3'), respectively. Solid (dashed) curves correspond to the ideal (noisy) model, where depolarizing noise with error rates of 10^{-5} and 10^{-4} is applied to one- and two-qubit gates. Each data point is obtained by averaging over 20 independent circuit executions, each consisting of 1000 measurement shots. The standard deviation across runs is at most 21 counts. Bidirectional arrows indicate the difference in the solution counts between the ideal and noisy models, evaluated at the optimal number of queries for each initialization scheme.

which enforces that each element is covered by exactly one selected subset.

Suppose that, after performing the classical preprocessing described in Algorithm 1 and reordering the qubit indices, we obtain a collection of disjoint subsets $C_1', \ldots, C_t'$ and reduced subsets $R_{t+1}', \ldots, R_s'$ derived from overlapping constraints. Let k_j denote the number of qubits in the j-th subset. For each disjoint subset C_j', we prepare the Dicke state $|D_1^{k_j}\rangle$ acting on the qubits $\mathbf{q}_{C_j'}$ to represent the corresponding constraint. For each reduced subset R_j', we employ the state $|D_{1,1}^{k_j}\rangle$ defined in Eq. (13), whose explicit form is

$$|D_{1,1}^{k_j}\rangle_{\mathbf{q}_{R_j'}} = \frac{1}{\sqrt{k_j+1}}|00\cdots0\rangle_{\mathbf{q}_{R_j'}} + \frac{\sqrt{k_j}}{\sqrt{k_j+1}}|D_1^{k_j}\rangle_{\mathbf{q}_{R_j'}}. \tag{45}$$

The state $|D_{1,1}^{k_j}\rangle$ can be prepared using the circuit shown in Figure 4.

We now look at the following example of the exact-cover problem. Let $U = \{1,2,3,4,5,6,7\}$ and the collection of subsets be given as

$$\begin{aligned} &A_1 = \{1,5\},\ A_2 = \{1,3,6\},\ A_3 = \{1,2,5\},\ A_4 = \{1,7\},\ A_5 = \{3,4\},\\ &A_6 = \{4,6\},\ A_7 = \{2,4,5\},\ A_8 = \{2,7\},\ A_9 = \{6,7\},\ A_{10} = \{3,5,7\}. \end{aligned} \tag{46}$$

This instance admits a unique exact-cover, given by $\mathscr{A}' = \{A_3, A_5, A_9\}$, so that $|S|$=1. From the subsets in Eq. (46), the corresponding constraint sets are constructed according to Eq. (43) as

$$\begin{aligned} &C_1 = \{1,2,3,4\},\ C_2 = \{3,7,8\},\ C_3 = \{2,5,10\},\ C_4 = \{5,6,7\},\\ &C_5 = \{1,3,7,10\},\ C_6 = \{2,6,9\},\ C_7 = \{4,8,9,10\}. \end{aligned} \tag{47}$$

Applying the classical preprocessing procedure described in Algorithm 1, we first identify disjoint sets that will be used to construct the circuit for initialization. When the threshold parameter $\eta_{\text{th}} = 0$, only strictly disjoint sets are allowed, and the preprocessing selects

$$C_1' = C_1,\ C_2' = C_4. \tag{48}$$

ARTICLE IN PRESS

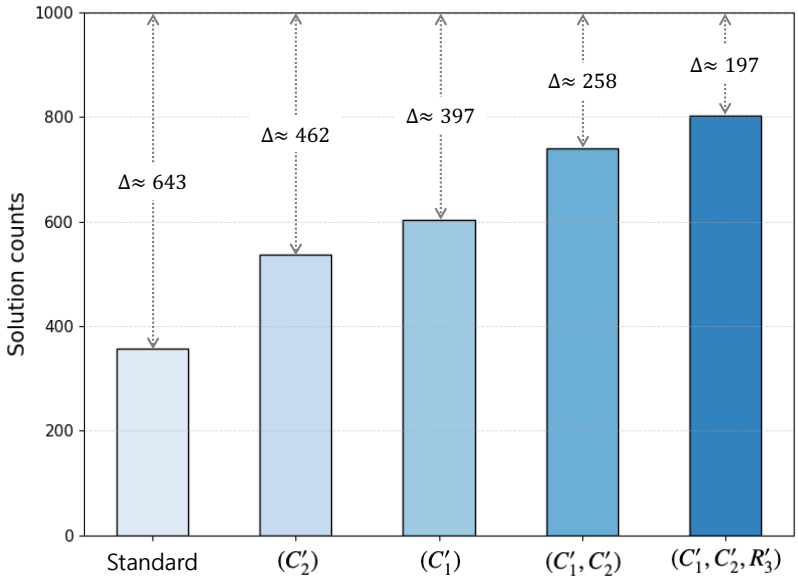


Figure 6. Difference in solution counts between the ideal and noisy models, evaluated at the respective optimal number of queries for different initialization strategies, where the bars show the solution counts in the noisy model and Δ denotes the corresponding difference. All simulation settings are the same as in Figure 5, and the standard deviation across runs is at most 19 counts. The strategies include the standard uniform initialization and several constraint-aware initializations incorporating single constraint sets (C_2') and (C_1'), as well as multiple constraint sets obtained through preprocessing, namely (C_1', C_2') and (C_1', C_2', R_3').

When the threshold is relaxed to $\eta_{\text{th}} = 1$, a small degree of overlap is permitted, which allows an additional set to be obtained as

$$R_3' = C_7 \setminus (C_1 \cup C_4) = \{8, 9, 10\}. \tag{49}$$

The initial quantum state is then constructed by preparing Dicke states on the subsystems corresponding to C_1' and C_2', while the subsystem associated with R_3' is prepared according to the quantum state of the form in Eq. (45). For comparison, the initialization strategies considered in this example, together with the corresponding initial states, search-space sizes, and optimal number of queries, are summarized in Table 2.

We perform simulations using IBM's Qiskit, an open-source software development kit for quantum circuit design and simulation[33]. The oracle operator is assumed to be ideal to reduce simulation overhead and to focus on the resource impact of the initialization and diffusion components. We note, however, that oracles for the exact-cover problem can be explicitly constructed using existing approaches, such as those proposed by Gilliam *et al.*[8] and Jiang *et al.*[26]. When simulating the noisy model, the multi-controlled X gate in the operator $2|0\rangle\langle 0|^{\otimes n} - \mathbf{I}$ is decomposed into one- and two-qubit gates over the gate set $\{RZ, H, CNOT\}$, consistent with the implementation used in the resource analysis. To assess the effect of noise on the performance of different initialization strategies, we consider a depolarizing noise model applied to all operations except for the oracle operator. Specifically, depolarizing noise with error rates of 10^{-5} and 10^{-4} is applied to one- and two-qubit gates, respectively. For each value of the query number, the circuit is executed 20 times, each run consisting of 1000 measurement shots. The reported results correspond to the average number of measurements yielding a solution state.

Figure 5 shows the solution counts as a function of the number of queries for the exact-cover instance defined in Eq. (46), obtained using the standard uniform initialization and two constraint-aware initialization strategies. The first constraint-aware strategy incorporates only the disjoint constraint sets (C_1', C_2'), while the second additionally includes the reduced set R_3' derived from overlapping constraints. For each initialization strategy, results are shown for both the ideal and noisy models. In the standard initialization, the corresponding optimal number of queries is 25, whereas the first constraint-aware strategy reduces the optimal number of queries to 7, and the second strategy further reduces it to 5. Although the constraint-aware initialization increases the circuit complexity associated with initial state preparation and the diffusion operator, the substantial reduction in the optimal number of queries leads to improved robustness against noise at the respective optimal query numbers.

Figure 6 extends the comparison by examining the difference in solution counts between the ideal and noisy models at the respective optimal number of queries for a broader set of initialization strategies. In addition to the standard uniform initialization and the two constraint-aware schemes based on (C_1', C_2') and (C_1', C_2', R_3'), which were discussed earlier, we also

ARTICLE IN PRESS

Strategy	Initial state	Search space size	Optimal query
Standard	$\|+\rangle^{\otimes 10}$	$2^{10} = 1024$	25
CA init. (C_2')	$\|D_1^3\rangle \otimes \|+\rangle^{\otimes 7}$	$3 \cdot 2^7 = 384$	15
CA init. (C_1')	$\|D_1^4\rangle \otimes \|+\rangle^{\otimes 6}$	$4 \cdot 2^6 = 256$	12
CA init. (C_1', C_2')	$\|D_1^4\rangle \otimes \|D_1^3\rangle \otimes \|+\rangle^{\otimes 3}$	$4 \cdot 3 \cdot 2^3 = 96$	7
CA init. (C_1', C_2', R_3')	$\|D_1^4\rangle \otimes \|D_1^3\rangle \otimes \|D_{1,1}^3\rangle$	$4 \cdot 3 \cdot 4 = 48$	5

Table 2. Summary of the initialization strategies used in the cardinality-constraint example. Here, CA init. $(\cdot)$ denotes the constraint-aware initialization obtained by incorporating the indicated constraint sets into the initial state preparation. The strategies CA init. (C_1', C_2') and CA init. (C_1', C_2', R_3') are obtained via the classical preprocessing in Algorithm 1.

include constraint-aware initializations incorporating only a single constraint set, namely (C_2') and (C_1'). These two cases correspond to reduced search spaces of size 3/8 and 1/4 of the reference space of size 2^{10}, respectively, which result in optimal query numbers of 15 and 12. As shown in Figure 6, the solution counts obtained in the noisy model increase as we move from the standard initialization to constraint-aware strategies that incorporate larger constraint sets and additional disjoint constraint sets. These results provide supporting evidence for the constraint selection strategy adopted in Algorithm 1, where larger constraint sets are prioritized and the procedure is designed to identify as many disjoint constraint sets as possible.

Importantly, Figure 6 shows that even in unfavorable preprocessing scenarios where only a single constraint set is identified, the corresponding constraint-aware initialization still yields a noticeable improvement over the standard uniform initialization. This suggests that the proposed preprocessing-guided initialization offers a systematic and practically feasible way to incorporate constraints. While incorporating additional constraints, including non-disjoint ones, may further improve performance, it is nontrivial to design a systematic initialization that directly incorporates non-disjoint constraint sets. Our preprocessing instead relaxes overlaps by extracting reduced subsets that can be implemented in the initialization, and the simulation results support the effectiveness of this approach. Moreover, identifying a large number of disjoint constraint sets is itself challenging. Accordingly, we adopt a greedy preprocessing strategy that does not aim to be optimal but provides a simple and implementable baseline that already delivers a clear performance advantage in our simulations.

Simulation results for parity constraints extracted from linear constraints

We now introduce a variant of the exact-cover problem to illustrate the incorporation of parity constraints extracted from linear constraints through GHZ-type initialization. The universal set U and the collection of subsets $\{A_i\}$ are taken to be the same as in the previous subsection. In contrast to the standard exact-cover formulation, we do not require the selected subsets to be disjoint. Instead, we require each element of U to be covered exactly twice. We assign weights α_i to the subsets such that when A_i is selected, it contributes α_i to the coverage count of each element it contains. We set $\alpha_i = 2$ for $i \in \{3, 6, 10\}$ and $\alpha_i = 1$ otherwise. Introducing binary variables $x_i \in \{0, 1\}$, this condition is expressed as

$$\sum_{i \in C_j} \alpha_i x_i = 2, \quad j = 1, \ldots, 7, \tag{50}$$

and this instance admits a unique solution given by $\mathscr{A}' = \{A_1, A_2, A_5, A_7, A_8, A_9\}$.

Using the definition of $\hat{C}_j = \{i \in C_j | \alpha_i \text{ is odd}\}$, we obtain

$$\hat{C}_1 = \{1,2,4\},\ \hat{C}_2 = \{7,8\},\ \hat{C}_3 = \{2,5\},\ \hat{C}_4 = \{5,7\},\ \hat{C}_5 = \{1,7\},\ \hat{C}_6 = \{2,9\},\ \hat{C}_7 = \{4,8,9\}. \tag{51}$$

Accordingly, by applying the preprocessing step in Algorithm 2, we identify disjoint sets $C_1' = \hat{C}_2$ and $C_2' = \hat{C}_3$. We then implement the GHZ-type initialization on the qubit registers associated with these constraint sets. Since two disjoint parity constraint sets are incorporated, the effective search space is reduced by a factor of $2^2 = 4$, which yields an optimal number of queries of 12 for this instance. Table 3 summarizes the constraint sets encoded by the proposed initialization strategy for this example, together with the corresponding initial states, search-space sizes, and optimal number of queries.

Figure 7 presents the simulation results obtained under the same settings as in the previous subsection. We plot the solution counts as a function of the number of queries for the ideal and noisy cases. The results compare three initialization strategies that include applying GHZ-type initialization to both C_1' and C_2', applying it only to C_1', and the standard uniform initialization. As in the cardinality case, incorporating additional constraint information leads to improved performance under noise. Even incorporating a single parity constraint set already yields a clear improvement over the standard uniform initialization. Together with the results in the previous subsection, this confirms that the proposed constraint-aware initialization framework can systematically exploit different types of linear constraints to improve practical performance.

ARTICLE IN PRESS

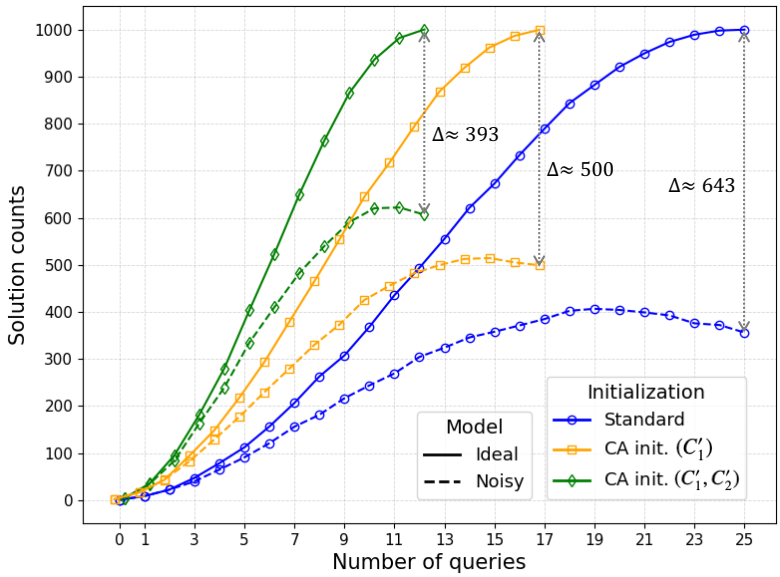


Figure 7. Comparison of the solution counts under different initialization strategies for parity constraints extracted from linear constraints. The standard uniform initialization is compared with two GHZ-type initializations incorporating (C_1') and (C_1', C_2'), respectively. Consistent with the cardinality case, incorporating additional parity constraint information improves performance under noise, and even incorporating a single constraint set yields a clear advantage over the uniform initialization. The simulation setting is the same as in Figure 5, and the standard deviation across runs is at most 21 counts.

Strategy	Initial state	Search space size	Optimal query
Standard	$\|+\rangle^{\otimes 10}$	$2^{10} = 1024$	25
CA init. (C_1')	$\|GHZ_{2,0}^{(X)}\rangle \otimes \|+\rangle^{\otimes 8}$	$2^9 = 512$	17
CA init. (C_1', C_2')	$\|GHZ_{2,0}^{(X)}\rangle \otimes \|GHZ_{2,0}^{(X)}\rangle \otimes \|+\rangle^{\otimes 6}$	$2^8 = 256$	12

Table 3. Summary of the initialization strategies used in the parity-constraints example. The strategy CA init. (C_1', C_2') is obtained via the classical preprocessing in Algorithm 2.

Conclusion

In this work, we presented a systematic framework for constraint-aware initialization in Grover's algorithm applied to combinatorial problems with linear constraints, which reduces the effective search space by incorporating constraint information already at the initialization step. The proposed approach combines a simple classical preprocessing step with structured initial state preparation. The preprocessing selects collections of mutually disjoint constraint sets that are jointly implementable in the initialization. For each selected set, we prepare a structured quantum state that encodes the corresponding constraint information. We use Dicke states to encode cardinality constraints and GHZ-type states to incorporate parity constraints extracted from more general linear constraints. The overall initial state is then constructed as a tensor product over the selected sets. We also provided a conservative circuit-level resource analysis, showing that although structured initialization introduces additional preparation costs, it can still improve overall resource efficiency in terms of gate counts and circuit depth compared to the standard uniform initialization. The practical effectiveness of the framework was further supported by numerical simulations of the exact-cover problem, demonstrating clear performance advantages, particularly in the presence of noise.

Several important directions remain for future work. While the present analysis focused on specific instances of Dicke states and GHZ-type states, more general Dicke states are known to admit polynomial-time circuit constructions[29], suggesting that similar resource-efficiency trends may extend beyond the cases considered here. In addition, the circuit implementations examined in this work are not optimized for minimal resource usage, and further improvements may be achieved through more efficient state preparation schemes. It would also be interesting to evaluate the proposed framework on real quantum hardware, where the impact of the platform-specific gate sets and noise characteristics on its practical resource efficiency could be assessed. The classical preprocessing procedure introduced in this framework is likewise intended as a simple and practical

ARTICLE IN PRESS

baseline rather than an optimal approach, and identifying more effective preprocessing methods will be important for achieving both practical efficiency and near-optimal performance.

Another natural direction is to extend the framework beyond equality-type linear constraints to inequality constraints. Such constraints may be incorporated by preparing superpositions of Dicke states across a range of Hamming weights, which suggests that the proposed approach could be generalized to a broader class of combinatorial problems. It would also be worthwhile to compare the proposed approach with divide-and-conquer strategies, in which the problem is decomposed into subproblems each solved by an independent algorithm. While such strategies reduce the per-instance resource requirements, they require either multiple quantum processors or sequential executions, whereas the proposed framework addresses the original problem within a single Grover search instance by incorporating available constraint information into the initialization. A concrete analysis of this trade-off remains an interesting open question. Finally, it would be interesting to apply the proposed framework within optimization settings such as Grover adaptive search[8] and variational algorithms with the Grover mixer quantum alternating operator ansatz[14], and to evaluate its impact on practically relevant combinatorial optimization problems. A thorough comparison of the proposed framework with related constraint-aware approaches in terms of resource efficiency and practical performance would also provide valuable insight.

Funding

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) (2019-0-00003) and the Global TOP strategic Research Program of the National Research Council of Science & Technology (NST) (GTL25011-000) grant funded by the Korea government (MSIT). M.C. acknowledges support from the Korea Institute of Science and Technology Information (KISTI) (Grant No. K26L3M3C3). E.B. is supported by the National Research Foundation of Korea (NRF) of Korea grant funded by the Ministry of Science and ICT (MSIT) (Grant No. RS-2025-07742969).

Code availability

The simulation code used in this work is publicly available at https://github.com/mathcmj/grover-ca-init and has been archived on Zenodo at https://doi.org/10.5281/zenodo.21385709.

References

1. Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 212–219 (1996).
2. Brassard, G., Hoyer, P., Mosca, M. & Tapp, A. Quantum amplitude amplification and estimation. *arXiv preprint quant-ph/0005055* (2000).
3. Montanaro, A. Quantum algorithms: an overview. *npj Quantum Inf.* **2**, 1–8 (2016).
4. Brassard, G., Høyer, P. & Tapp, A. Quantum counting. In *International Colloquium on Automata, Languages, and Programming*, 820–831 (Springer, 1998).
5. Aaronson, S. & Rall, P. Quantum approximate counting, simplified. In *Symposium on simplicity in algorithms*, 24–32 (SIAM, 2020).
6. Durr, C. & Høyer, P. A quantum algorithm for finding the minimum. *arXiv preprint arXiv:quant-ph/9607014* (1996).
7. Baritompa, W. P., Bulger, D. W. & Wood, G. R. Grover's quantum algorithm applied to global optimization. *SIAM J. on Optim.* **15**, 1170–1184 (2005).
8. Gilliam, A., Woerner, S. & Gonciulea, C. Grover adaptive search for constrained polynomial binary optimization. *Quantum* **5**, 428 (2021).
9. Nigatu, H. *et al.* Quantum machine learning and grover's algorithm for quantum optimization of robotic manipulators. *IEEE Robotics Autom. Lett.* **10**, 13090–13097 (2025).
10. Campbell, E., Khurana, A. & Montanaro, A. Applying quantum algorithms to constraint satisfaction problems. *Quantum* **3**, 167 (2019).
11. Ohno, H. Grover's search with learning oracle for constrained binary optimization problems. *Quantum Mach. Intell.* **6**, 12 (2024).
12. Sano, Y., Mitarai, K., Yamamoto, N. & Ishikawa, N. Accelerating grover adaptive search: Qubit and gate count reduction strategies with higher order formulations. *IEEE Transactions on Quantum Eng.* **5**, 1–12 (2024).

ARTICLE IN PRESS

13. Hadfield, S. *et al.* From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12** (2019).

14. Bärtschi, A. & Eidenbenz, S. Grover mixers for qaoa: Shifting complexity from mixer design to state preparation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 72–82 (IEEE, 2020).

15. Vikstål, P. *et al.* Applying the quantum approximate optimization algorithm to the tail-assignment problem. *Phys. Rev. Appl.* **14**, 034009 (2020).

16. Zhang, Z. *et al.* Grover-qaoa for 3-sat: quadratic speedup, fair-sampling, and parameter clustering. *Quantum Sci. Technol.* **10**, 015022 (2025).

17. Bengtsson, A. *et al.* Improved success probability with greater circuit depth for the quantum approximate optimization algorithm. *Phys. Rev. Appl.* **14**, 034010 (2020).

18. Metwalli, S. A., Le Gall, F. & Van Meter, R. Finding small and large k-clique instances on a quantum computer. *IEEE Transactions on Quantum Eng.* **1**, 1–11 (2020).

19. Mikuriya, T., Yukiyoshi, K., Fujiwara, S., de Abreu, G. T. F. & Ishikawa, N. Grover adaptive search for the higher-order formulation of quadratic assignment problems. *arXiv preprint arXiv:2410.12181* (2024).

20. Griset, R., Lavdas, I. & Guth Jarkovský, J. State-space reduction techniques exploiting specific constraints for quantum search initialization, application to an outage planning problem. *J. Quantum Comput.* **7**, 81–105 (2025).

21. Wilkening, S., Ziegler, T. & Hess, M. Constraint-oriented biased quantum search for linear constrained combinatorial optimization problems. *arXiv preprint arXiv:2512.05205* (2025).

22. Cook, J., Eidenbenz, S. & Bärtschi, A. The quantum alternating operator ansatz on maximum k-vertex cover. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 83–92 (2020).

23. Christiansen, P., Binkowski, L., Ramacciotti, D. & Wilkening, S. Quantum tree generator improves qaoa state-of-the-art for the knapsack problem. In *Proceedings of the IEEE International Conference on Quantum Computing and Engineering (QCE 2025)*, 1–10 (2025).

24. Dicke, R. H. Coherence in spontaneous radiation processes. *Phys. Rev.* **93**, 99 (1954).

25. Greenberger, D. M., Horne, M. A. & Zeilinger, A. Going beyond bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*, 69–72 (Springer, 1989).

26. Jiang, J.-R. & Wang, Y.-J. Quantum circuit based on grover's algorithm to solve exact cover problem. In *2023 VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 1–5 (IEEE, 2023).

27. Sato, R., Saito, K., Nikuni, T. & Watabe, S. Embedding all feasible solutions of traveling salesman problem by divide-and-conquer quantum search. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 2, 270–271 (IEEE, 2023).

28. Sato, R. *et al.* Two-step quantum search algorithm for solving traveling salesman problems. *IEEE Transactions on Quantum Eng.* (2025).

29. Bärtschi, A. & Eidenbenz, S. Deterministic preparation of dicke states. In *International Symposium on Fundamentals of Computation Theory*, 126–139 (Springer, 2019).

30. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).

31. Maslov, D. Advantages of using relative-phase toffoli gates with an application to multiple control toffoli optimization. *Phys. Rev. A* **93**, 022311 (2016).

32. Wong, T. G. *Introduction to classical and quantum computing* (Rooted Grove Omaha, NE, USA, 2022).

33. Javadi-Abhari, A. *et al.* Quantum computing with qiskit. *arXiv preprint arXiv:2405.08810* (2024).

Author contributions statement

E.B. and M.C. conceived the idea and performed the calculations. M.C. carried out the simulations. All authors discussed the results and contributed to the preparation of the manuscript.

Competing interests

The authors declare no competing interests.

ARTICLE IN PRESS

Additional information

Correspondence and requests for materials should be addressed to M.C.

ARTICLE IN PRESS